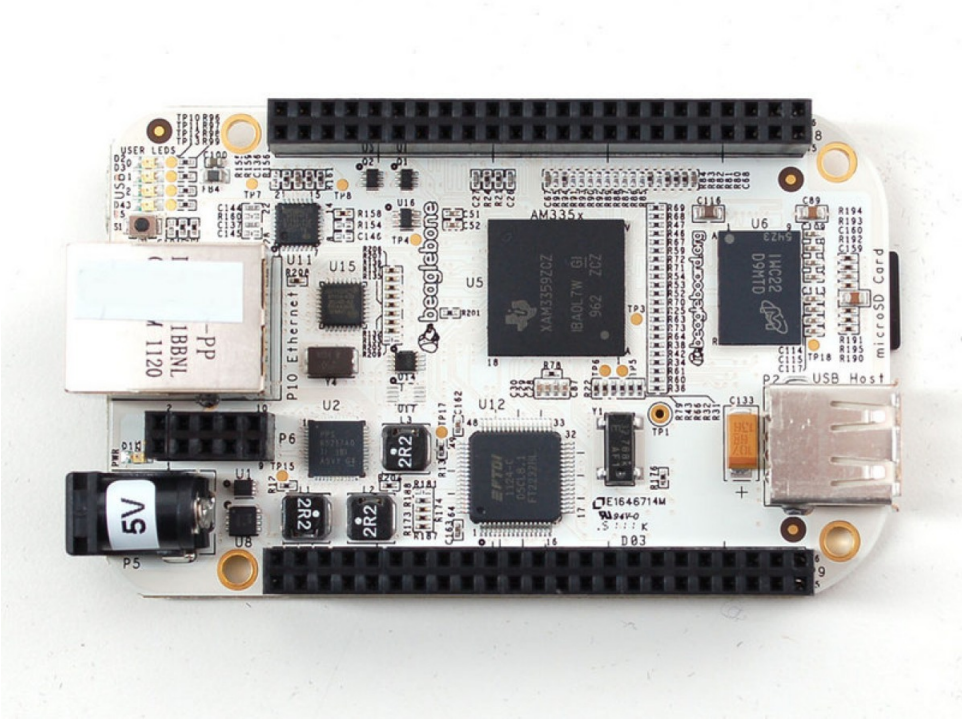
beaglebone



Beaglebone Guide: GPIO programming on the Beaglebone

Jayneil Dalal(jayneil.dalal@gmail.com)

February 7, 2013

Abstract

In this guide, I will describe how to program a GPIO on the Beaglebone to toggle LED step by step. This guide targets beginners who are just getting started on the Beaglebone.

Specifications

Processor

- 720MHz super-scalar ARM Cortex-A8 (armv7a)
- 3D graphics accelerator ARM Cortex-M3 for power management
- 2x Programmable Realtime Unit 32-bit RISC CPUs

Connectivity

- USB client: power, debug and device
- USB host
- Ethernet
- 2x 46 pin headers 2x I2C, 5x UART, I2S, SPI, CAN, 66x 3.3V GPIO, 7x ADC

Software

- 4GB microSD card with Angstrom Distribution
- Cloud9 IDE on Node.JS with Bonescript library

Contents of the box

When you purchase a Beaglebone, you get the following items in the box as shown in Figure -1:

1. Beaglebone
2. USB cable
3. 4gb micro sd card

Figure-1: Box contents

Expansion Headers on the Beaglebone

The expansion headers on the beaglebone are comprised of two 46 pin connectors which are P8 and P9. All signals on the expansion headers are 3.3V unless otherwise indicated.

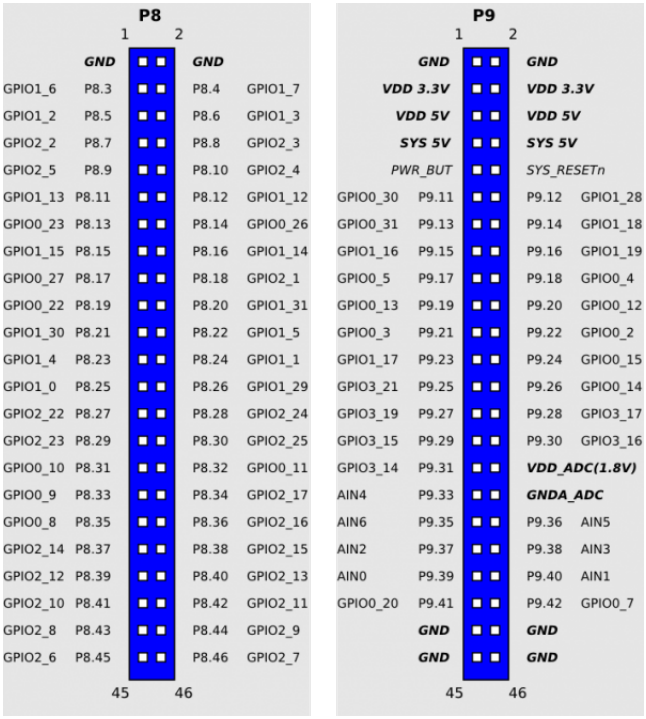


Figure-2: *Expansion headers on the Beaglebone*

Selecting the correct LED

- To make sure that you do not damage the GPIO pins on the Beaglebone, please use a LED whose rating should not exceed 3.3V/6mA:

  http://www.digikey.com/product-detail/en/HLMP-4700-C0002/516-2483-2-ND/1234840

- If you have an LED that is higher than the above mentioned ratings, please refer the refer the link below to select an appropriate current limiting resistor:

  http://www.cmiyc.com/tutorials/led-basics/

GPIO on the Beaglebone

The pins on the expansion header have multiple functions. To find out what is the default function of a pin, refer the beaglebone reference manual which can be downloaded from the link below:

http://beagleboard.org/static/BONESRM_latest.pdf

For example, Table-8 on page - 54 describes the default function of each pin on P8 expansion header under the '**SIGNAL NAME**' column. The '**CONN**' column describes the actual pin number as seen on the physical board. Tables-9,10 list the other possible functions of a particular pin on the P8 expansion header.

Once you have identified the pin number which you would like to use as a GPIO, you need to find out its corresponding reference number in the kernel. For example, if you would like to use pin 23 on P8 expansion header, then find out its default function as mentioned earlier. Note down the entire signal name. In this case, pin 23 is **GPIO1_4**. So any GPIO you come across would be referenced as GPIOM_N. Identify M,N. Use the formula below to find the corresponding reference number in the kernel:

Reference number = Mx32 + Y

Hence, pin 23 would be referenced as gpio 36 in the kernel.

Now, to change the function of a pin using the kernel you need to access the /sys/kernel/debug/omap_mux directory via the terminal on the beaglebone. Here your pin will be referenced by the name it is assigned in its mode 0. So, in table - 9, pin 23 is referenced as **gpmc_ad4** in mode 0. Then, identify the mode in which the pin can be used as GPIO. For pin 23, the mode is 7.

Preparing the SD Card

The Beaglebone already comes with an sd card that is preloaded with a working Angstrom image. In any case should you want a newer image or want to program the sd card again, this section covers it all.

- First download the latest Angstrom image for Beaglebone from the link below:

  http://downloads.angstrom-distribution.org/demo/beaglebone/

  At the time of writing this guide, the latest image available for download was 'Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.05-beaglebone-2012.11.22.img.xz'

- Now, identify the correct raw device name (like /dev/sde - not /dev/sde1) for the sd card

- Now unpack the image to the sd card by writing the following command in the terminal:

  ```
  $ xz -dkc Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.05-beaglebone-2012.11.22.img.xz > /dev/sdX
  ```

  Here 'sdX' stands for the device id of the sd card.

Setup

Please refer the figure below to see the connections:

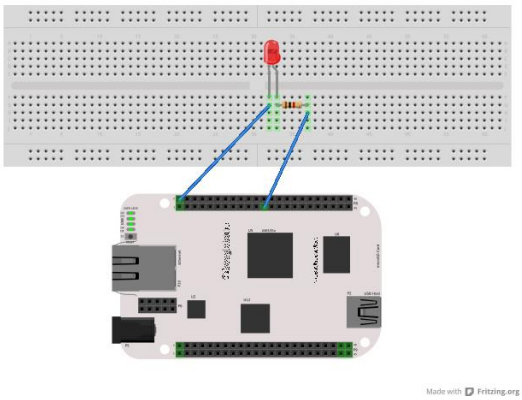


Figure-3: Connection Diagram

Only the pins on the P8 expansion header are used in this case. Connect the anode of the LED to the resistor(I am using a 110 ohms resistor) which in turn is connected to pin 23(GPIO) and connect the cathode of the LED to pin 2(GND) .

Steps

For the purpose of this guide, I have used an Ubuntu 12.04 64 bit system.

- To power up the beaglebone, connect it to the computer via the usb cable as shown in the Figure -4:

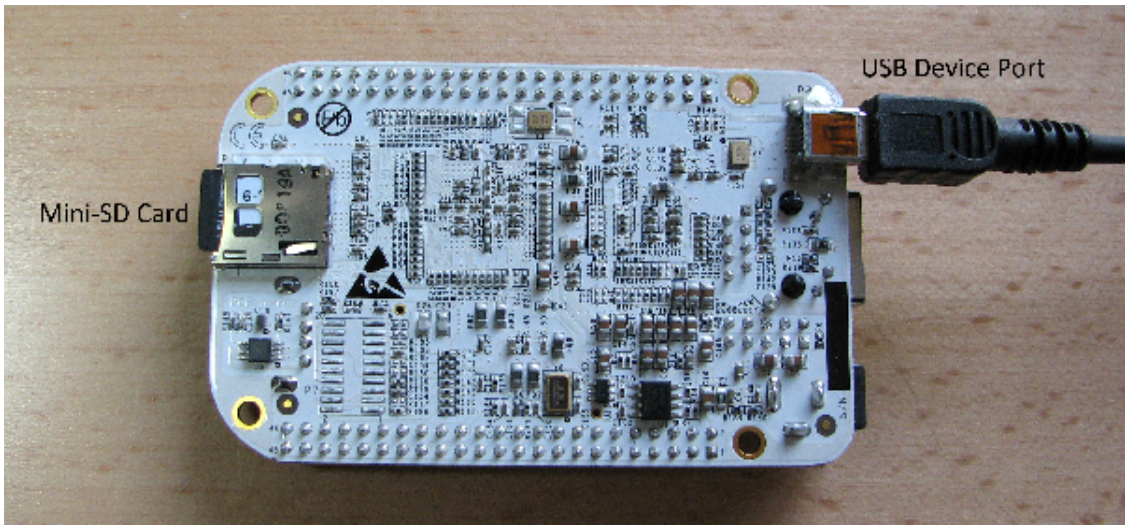


Figure-4: Powering up the Beaglebone

- Eject the Beaglebone via the disk utility program in Ubuntu. I tried ejecting it via the file manager but that did not work for me. Upon every boot, the Beaglebone is the "storage mode" by default. Hence, this step is done to switch it to "network mode".
- Access the beaglebone via the terminal:

```
$ screen /dev/ttyUSB1 115200
```

Note:- You can also use minicom. But this is just much easier! Also in most cases the virtual USB serial port is ttyUSB1. If it does not work, try ttyUSB0 .

- You should be greeted by an Angstrom login. The username for the same is 'root' and for password, just press 'ENTER'. You should see the following prompt:

```
root@beaglebone:~#
```

- Make sure that your kernel supports GPIO by typing the following commands in the terminal:

```
$ grep GPIOLIB /boot/config-‘uname -r‘
```

The output after running the above command should be as shown below:

```
CONFIG_ARCH_REQUIRE_GPIOLIB=y
CONFIG_GPIOLIB=y
```

Now run the following command in the terminal:

```
$ grep GPIO_SYSFS /boot/config-‘uname -r‘
```

The output after running the above command should be as shown below:

```
CONFIG_GPIO_SYSFS=y
```

- Make sure that the pin you are using is configured to be in the gpio mode. So, run the following command in the terminal:

```
echo 7 > /sys/kernel/debug/omap_mux/gpmc_ad4
```

- Export the pin

```
echo 36 > /sys/class/gpio/export
```

- Now let us light that LED!

Since its a GPIO, we need to configure it in the output mode. Write the following commands in your terminal

```
echo out > /sys/class/gpio/gpio32/direction
```

Now, let us toggle the LED by typing the following commands in terminal (First one is for turning ON and latter for OFF):

```
echo 1 > /sys/class/gpio/gpio32/value
echo 0 > /sys/class/gpio/gpio32/value
```

- Unexport the pin once finished

```
echo 36 > /sys/class/gpio/unexport
```

You should get the output as shown below:

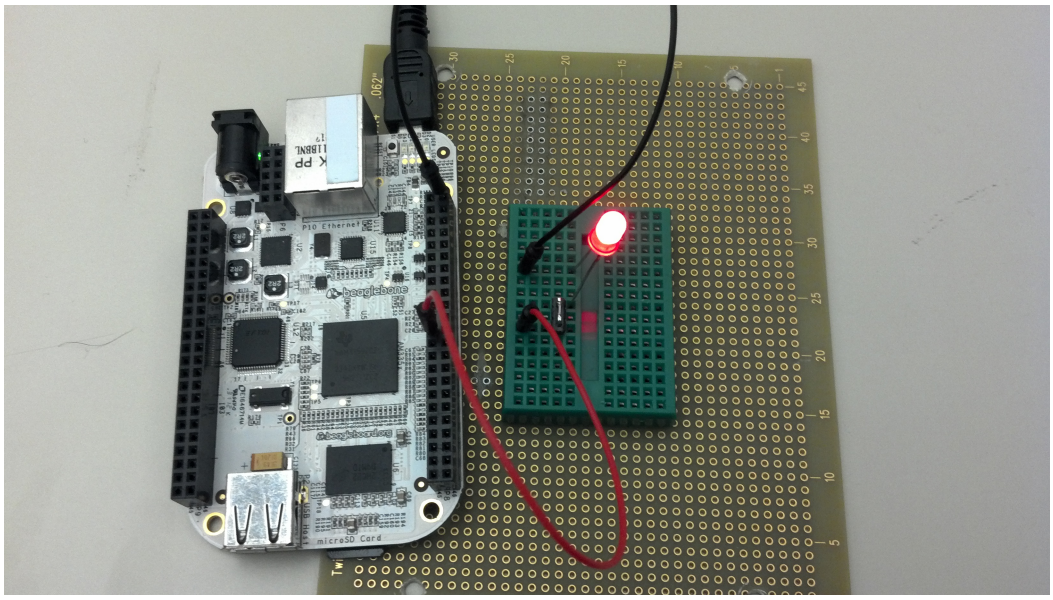


Figure-5: Led ON

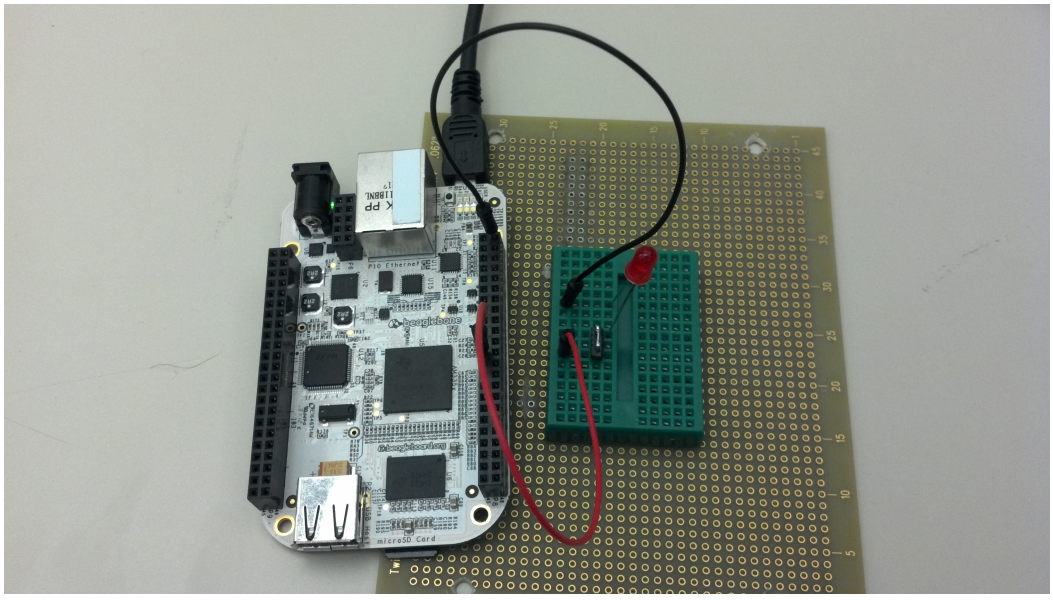


Figure-6: Led OFF